

TESTE DE DESENVOLVIMENTO DE SISTEMAS - PYTHON

Nome Completo: André Trindade Raposo

Data: 19 de agosto de 2020

Instrução Inicial: Crie na área de Trabalho, uma nova pasta e renomeie a mesma com seu nome. Você utilizara esta pasta para salvar os testes abaixo.

1. Nenhum corpo que não seja sólido é cristal. Conclui-se logicamente que:
(Somente uma resposta correta)

☐ Algum cristal não é corpo sólido
☐ Todo corpo sólido é cristal
☒ **Todo cristal é corpo sólido**
☐ Nenhum corpo sólido é cristal
☐ Nenhum cristal é corpo sólido

2. A soma de dois números é 16 e a diferença dos seus quadrados é 128. O produto desses dois números é:
(Somente uma resposta correta)

☐ -36
☐ -12
☐ 12
☐ 36
☒ **48**

3. Em determinada rua, há 5 residências que estão numeradas em forma de progressão geométrica crescente, de modo que a primeira residência recebeu numeração 3, e a segunda, numeração 12. Com base nessas informações, assinalar a alternativa CORRETA:
(Somente uma resposta correta)

☐ A numeração da terceira casa é 21.
☐ A numeração da quarta casa é 30.
☐ Somando-se a numeração da primeira e da terceira casa, o resultado é o número da quarta casa.
☒ **O resultado da soma da numeração da terceira e da quarta casa é igual a 240.**
☐ O resultado da soma da numeração das cinco casas é igual a 1.025.

4. Se:
1=6
2=26
3=326
4=4.326
5= ?
Qual será o valor correspondente ao 5? **54.326**

5. Um método se torna multithread quando é declarado como Async e esperado com await?

(Somente uma resposta correta)

() Sim, Async/Await fazem o método ser executado em uma nova thread.

() Sim, Async/Await tornam opcional a criação de novas threads.

(x) Não, Async/Await nunca é executado em threads diferentes da onde foram criados

() Não, Async/Await não é uma garantia que o método ira ser executado em uma nova thread.

() Sim, Async/Await forçam o método a executar de forma assíncrona e concorrente.

6. Usar programação assíncrona (Async/Await) garante que a execução do código seja mais rápida:

(Somente uma resposta correta)

() sim, assíncrono sempre tem performance melhor que síncrono.

() Não, toda programação assíncrona carrega um grande overhead tornando-a lenta em comparação com síncrona.

() Se a operação assíncrona for um código muito pequeno, o overhead do assíncrono pode afetar positivamente a performance.

() Cada método executa em uma velocidade diferente, e a escolha entre síncrono e assíncrono não afeta a performance do código.

(x) A programação assíncrona executa tarefas em paralelo, o código pode terminar a execução mais rápido mesmo sem mudança de performance.

7. Qual o resultado do programa a seguir?

(Somente uma resposta correta)

```
internal class Program
{
    private static void Main(string[] args)
    {
        int? n1 = null;
        int? n2 = 45;
        int? n3 = null;
```

```

        Console.WriteLine(n1 > n2);
        Console.WriteLine(n1 < n2);
        Console.WriteLine(n1 != n2);
        Console.WriteLine(n1 == n2);
        Console.WriteLine(n1 == n3);
    }
}
( ) False, False, False, False, False,
( ) False, True, True, False, True
( ) True, False, True, False, True
(x) False, False, True, False, True
( ) True, True, False, False, True

```

8. Escrever um código utilizando português estruturado ou Pseudocódigo, no qual o resultado final do Vetor deve conter os mesmos números ordenados crescentemente. Considerando o seguinte vetor = {5,3,2,4,7,1,0,6}
(Não utilize funções de ordenação prontas)

```

FUNÇÃO ORDENAÇÃO CRESCENTE (VETOR [], TAMANHO) :
    I = 1

    // LOOP PARA PERCORRER O VETOR
    ENQUANTO I < (TAMANHO) :
        J = I - 1

        // LOOP PARA SUBSTITUIR OS VALORES
        ENQUANTO J >= 0 E VETOR[J] > VETOR[J+1] :
            AUXILIAR = VETOR[J]
            VETOR[J] = VETOR[J+1]
            VETOR[J+1] = AUXILIAR
            J = J - 1

        I = I + 1

```

9. Desenvolvimento deverá ocorrer com o back-end em Python e o banco de dados preferencialmente em Mongo. Após finalizar o desenvolvimento subir no github o código fonte e banco de dados.

Você deverá criar uma API para um portal de notícias para cadastro de notícias, pesquisa de notícias e visualização de notícias.

No cadastro de notícias o usuário poderá informar os seguintes dados:

- Título da notícia (obrigatório);
- Texto da notícia (ilimitado e obrigatório);

- Autor (chave estrangeira para a tabela Autor e é obrigatório).

Também devem existir a opção de editar e excluir.

Na pesquisa de notícias o usuário poderá pesquisar pelas notícias cadastradas no banco de dados. A consulta ocorrerá somente por um parâmetro. A consulta à tabela de notícias deve ser feita nos campos título, texto e nome do autor.

Para visualizar notícias realizar a busca no banco de dados de todos os campos da notícia e realizar a listagem destas.

Você deverá criar um banco de dados, preferencialmente em Mongo, você pode definir a estrutura.

Os arquivos estão dentro da pasta 'ex_9', dentro contém 3 outras pastas:

- "database", que contém duas coleções de dados com extensão .json, que podem ser importados para o banco de dados MongoDB
- "scr", contém os arquivos do projeto e um arquivo para instalação de dependências da venv "requirements.txt".
- "postman": contém uma coleção de requests com extensão .json, que pode ser importado para o Postman para facilitar os testes.

O banco de dados que o programa procura ou cria caso não exista é chamado de 'noticias' URLs:

localhost:5000/noticias/ - [GET e POST]
POST envia como forms os seguintes campos (titulo, conteudo, autor, email, username)

localhost:5000/noticia/id - [GET, PUT e DELETE]
id é uma int e se inicia a partir de 1.
PUT envia como forms os seguintes campos (titulo, conteudo)

localhost:5000/noticia/?arg=filtro - [GET]
filtro é uma string
GET envia como argumento apenas um campo, que pode ser (titulo, conteudo ou autor)